WHAT IS CLAIMED IS:

1. A method, comprising:

determining that a first software component is to use a device, the device being shared with a second software component; and

arranging via information stored at the device to use the device when the second software component is not using the device.

2. The method of claim 1, wherein the information stored at the device includes:

a first component flag,

a second component flag, and

a turn flag,

wherein the first and second component flags can indicate free or busy and the turn flag can indicate first component or second component.

3. The method of claim 2, wherein the first and second component flags are initialized to free.

4. The method of claim 3, wherein said arranging further comprises:

setting the first component flag to busy; and

setting the turn flag to second component.

5. The method of claim 4, wherein said arranging further comprises:

waiting until either the second component flag is free or the turn flag is first component before using the device.

6. The method of claim 5, wherein said arranging further comprises:

setting the first component flag to free after using the device.

7. The method of claim 1, wherein said arranging is performed in accordance with at least one of: (i) Dekker's algorithm, and (ii) Peterson's algorithm.

8. The method of claim 1, wherein the information is stored using pre-fetchable storage registers at the device.

9. The method of claim 1, wherein at least one of the first and second software components comprise a device driver.

10. The method of claim 1, wherein the device comprises at least one of: (i) a network adapter, and (ii) a disk controller.

11. The method of claim 1, wherein the device comprises a network controller, the first software component comprises a network driver, and the second software component comprises an encryption driver.

12. The method of claim 1, wherein the first and second software components access the device via at least one of (i) a peripheral component interconnect bus and (ii) a universal serial bus.

13. An apparatus, comprising:

a storage medium having stored thereon instructions that when executed by a machine result in the following:

determining that a first software component is to use a device, the device being shared with a second software component, and

arranging via information stored at the device to use the device when the second software component is not using the device.

14. The apparatus of claim 13, wherein the information stored at the device comprises:

a first component flag,

a second component flag, and

a turn flag,

wherein the first and second component flags can indicate free or busy and the turn flag can indicate first component or second component.

15. The apparatus of claim 14, wherein the first and second component flags are initialized to free.

16. The apparatus of claim 15, wherein said arranging further comprises:

setting the first component flag to busy, and

setting the turn flag to second component.

17. The apparatus of claim 16, wherein said arranging further comprises:

waiting until either the second component flag is free or the turn flag is first component one before using the device.

18. The apparatus of claim 17, wherein said arranging further comprises:

setting the first component flag to free after using the device.

19. The apparatus of claim 13, wherein said arranging is performed in accordance with at least one of: (i) Dekker's algorithm, and (ii) Peterson's algorithm.

20. The apparatus of claim 13, wherein the information is stored using pre-fetchable storage registers at the device.

21. The apparatus of claim 13, wherein at least one of the first and second software components comprise a device driver.

22. The apparatus of claim 13, wherein the device comprises at least one of: (i) a network adapter, and (ii) a disk controller.

23. An apparatus, comprising:

a processor; and

a bus interface,

wherein information stored at a device accessed via the bus interface is to arrange for a first software component of the processor to use the device when a second software component of the processor is not using the device.

24. The apparatus of claim 23, wherein the device comprises a peripheral device and the information is stored using pre-fetchable registers at the peripheral device.

25. A system, comprising:

a processor;

a network adapter; and

an Ethernet port coupled to the network adapter,

wherein information stored at the network adapter is to arrange for a first software component of the processor to use the network adapter when a second software component of the processor is not using the network adapter.

26. The system of claim 25, wherein the first software component comprises a network driver and the second software component comprises an encryption driver.